

アセンブリ言語トレーニングのための マルチマイクロコンピュータシステム

柳瀬龍郎* 松山幸雄* 篠 競* 安達治彦* 水野広治* 鈴木重寛* 奥村彰二*

Multi-Microcomputer System for Training of Assembly Language Programming

Tatsurou YANASE, Yukio MATSUYAMA, Kisou SHINO, Haruhiko ADACHI,
Hiroharu MIZUNO, Shigehiro SUZUKI, and Shoji OKUMURA

(Received Aug. 25, 1990)

We develop a multi-microcomputer system, which is aimed to be used for student laboratory of assembly language programming in our department of computer science. It may be expensive and inconvenient to set up in a class so many single board computers as every student can use one of them. A CPU board in the system of a size of the standard Europac board (233.35×220 mm) has 4 microprocessors, and each of them has local memories and two serial line interface channels so that it can run independently as a single board computer. The system has following features:

- 1) Shared common memories, with local memories for each processor.
- 2) Shared ROM for system monitor.
- 3) The monitor has basic functions such as program debugging, terminal I/O for user interface and communication I/O for program loading from other computer systems.

The performance of the system used as a multi-microprocessor system is evaluated for a few parallel programs.

1. はじめに

著者らの所属する学科（福井大学工学部情報工学科）におけるカリキュラムには講義に伴った時間外の演習の他に、計算機を使ったプログラミングの演習がいくつか組み入れているが、その内容は以下のようなものである。

- 1) 大型計算機を使った構造化プログラミングの演習（1年生前期）
- 2) アセンブリ言語のプログラミング演習（2年生前期）
- 3) UNIXワークステーションでのC言語のプログラミング演習（2年生後期）
- 4) PASCAL言語のプログラミング演習（3年生後期）

このうち、2)は68000 マイクロプロセッサを対象とした、アセンブリ言語のプログラミング演習であるが、そのプログラムの実行のターゲットマシンとして、コンピュータを設計製作した。本報告は、このコンピュータの構成と機能およびその特徴などについて述べたものである。図1に、この演習で使用するシステム全体の概略図を示す。

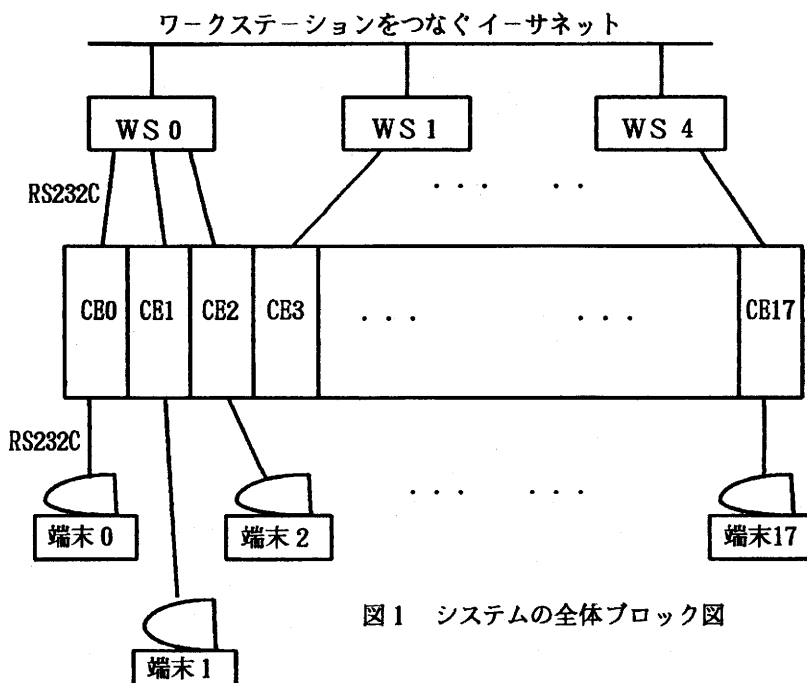


図1 システムの全体ブロック図

図中のWS0～WS4はUNIXワークステーション(LUNA)であり互いにイーサネットで接続されている。それぞれのワークステーションには、4ないし5本のシリアル回線により端末が接続されている。アセンブリ言語の演習を行うとき、それらの途中にコンピュータエレメント（以後これをCBと略記する）が一組接続される。CBは2個のRS232Cポートを備えており、簡単なモニタによって制御されている。そのモニタは2個の回線から入力されたコードをそれぞれ他の回線へそのまま送り出す機能を働かすことも可能であり、そのときは端末は単純にホストWSに直接接続する場合とまったく同じように機能する。学生は一台の端末の前に坐り、適当なコマンドを入力することによって、WSとCBの両方の端末として機能させることが出来る。学生は；

- 1) ワークステーションでアセンブリ言語のプログラムを作り、
 - 2) これをアセンブルしてオブジェクトプログラムにし、
 - 3) このオブジェクトプログラムをワークステーションからCBに転送して
 - 4) CBにおいてオブジェクトプログラムを実行、デバッグする
 - 5) ワークステーションでアセンブリ言語のプログラムを修正、保存する
- 以下、2)－5)を繰り返し、正しく動作するアセンブリプログラムを作り上げる。

2. システムの構成とハードウェア

ワークステーション(WS)はOMRON 社製 LUNAシリーズを使用し、OSはUNIXシステムV系を用いている。CBはWSと端末の間にあって、WSからみれば1つユーザであり、端末からみれば簡単なコマンドを受け付けるコンピュータとなっている。図2に今回制作したマルチコンピュータのブロック図を示す。CBは実際は1枚のボードに3組のCBが搭載されており、これが5枚、そのほかに1枚の共通ボードがあり、これら6枚のボードがバスによって接続されている。共有ボードにはバスアービタ、モニタROM、共有メモリなどが載っている。

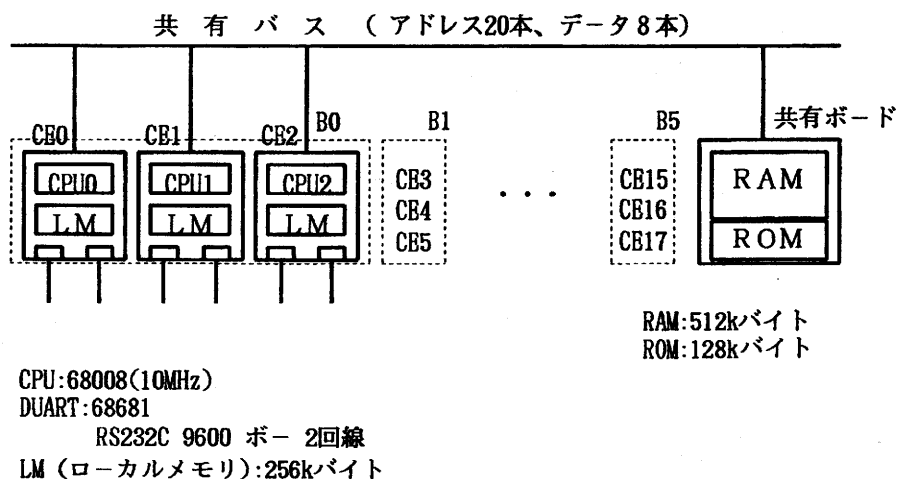


図2 マルチコンピュータのブロック図

システム全体としては次のような特徴を有する。

- 1) 分散共有型メモリ結合マルチマイクロコンピュータ
- 2) モニタROM は共有メモリ空間内に存在し、各CPU はRAM メモリのみをローカル空間内に持つ
- 3) モニタ機能は、コンソールに対するI/O、シリアル回線からのオブジェクトプログラムのダウンロード、トレースなど

3. ハードウェア

3.1 CEボード

CBは図3に示すように極めて単純であり、以下の要素で構成されている。

1) CPU

(モトローラ) マイクロプロセッサMC68008(10MHz)

[内部レジスタ 32ビット×16本、データバス8ビット、アドレスバス20ビット]

2) ローカルメモリ

(富士通) DRAMモジュールMB85225-12、8ビット×256kワード1個をDRAMC1422Aで制御

3) シリアルインターフェイス

(モトローラ) DUART MC68681PによりRS232C、2回線を制御

4) 共有バスバッファ

データバス 8 ビット、アドレスバス 20 ビットをバッファリング

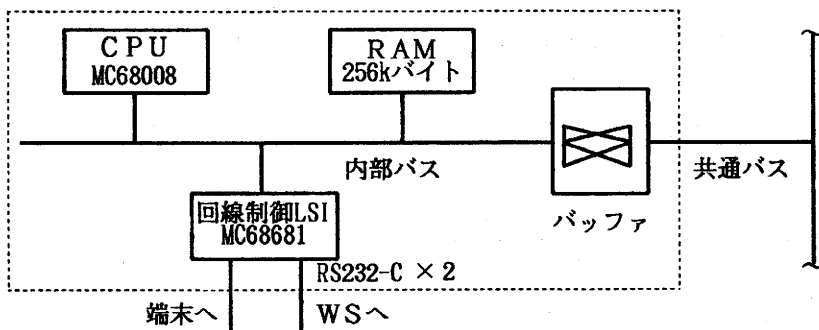


図3 CEのブロック図

3. 2 共有ボード

このマルチコンピュータの特徴でもあり、全てのCBから必ずアクセスされる重要な回路を構成している。

1) アービタ

24MHz のクロックで動作するリングアービタ

2) 共有メモリ回路

a) ROM

BPROM MB27C256A-20×5 (CEのモニタやその他のプログラムが格納されており、各CBはこれを直接実行したり、あるいはいったんローカルメモリにコピーしてこれを実行したりする)

b) RAM

(富士通) DRAMモジュールMB85225-12、8ビット×256kワード2個をDRAMC1422Aで制御
[マルチプロセッサとしてこのシステムを使用するとき、CB相互でデータを交換するために使用することが出来る]

3. 3 リセットメカニズム

個々のCEにはリセットスイッチが備えられており、CE毎にリセットが可能となっている。これは、学生がプログラムを走らせた後、それがリセットする必要が生じたとき、他のCEに影響を与えないでリセット出来るようにしたものである。CEのリセットボタンを押すことによって、そのCEのハードウェアがリセットされ、CPUからのリセットベクタの獲得のための0～7番地へのアクセスもROMに向かってなされる。リセットベクタはモニタのコピールーチンの先頭にセットされているが、見掛け上高位のアドレス部に設定されており、CPUがここをアクセスした直後にハード的な処理がなされ、これによって以後のCPUの低位アドレス部に対するアクセスは全てROMではなくローカルRAMに対してなされることになる。図4はこのメモリマップの変化を示している。リセットベクタ取得後のCPUの仕事は共有ROM内に格納されているモニタプログラムをROMから、CE内のローカル

RAM へコピーし、そのスタート番地へジャンプすることである。これによってCBは端末からのコマンドを受け入れる状態となる。

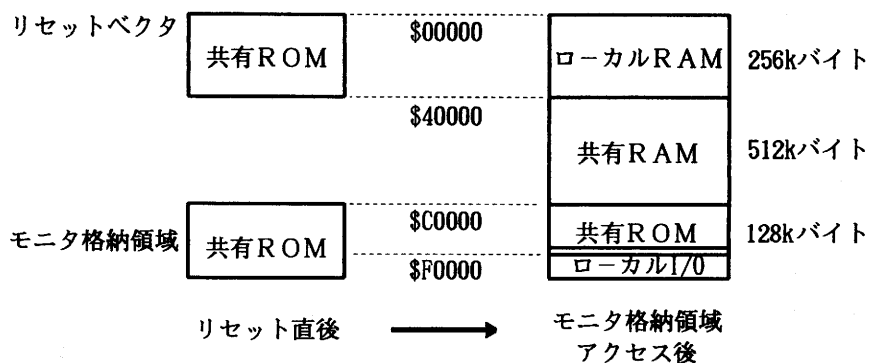


図4 メモリマップの変化

アドレス	用途	割当	種類
①\$0~\$3000-1 \$68	システム 割り込みベクタ	ローカル	RAM
②\$3000~\$38000-1	ユーザ		
③\$38000~\$40000-1	システム		
④\$40000~\$C0000-1	ユーザ	共有	RAM
⑤\$C0000~\$F0000-1 \$C8800~\$C8812 \$E8800~\$E8803	システム 割り込み(w) CPU 番号(R)		ROM
⑥\$F0000~\$100000-1 \$F0000~\$F000F \$F8000	システム DUART(R/W) 割り込みF/F リセット(R)		I/O レジスタ

表1 アドレスマップ

4. メモリ空間

CEのCPU から見たメモリマップを表1に示す。CEのローカル空間は自身のCPU からのみアクセス可能であるが、共有メモリ空間はどのCEからもアクセスが可能である。したがって、各CEはこの共有メモリを通じて、他の任意のCEとデータの交換、連絡が可能である。

5. モニタ

5. 1 特徴

通常の状態ではローカルメモリ空間の\$3C000～\$3FFFFに格納されているモニタが各CEの機能を制御する。前にも述べたように、これはリセット直後に共有メモリ空間のROM からコピーされたものである。すなわち、全てのCEは共有メモリ空間にあるただ一つのROM 上のモニタをコピーする訳である。したがって、各CEのモニタはすべて全く同じ内容のものとなる。このメカニズムのために個々のCEが各々のROM を個別に備える必要がなく、また改良したモニタを置き換える場合にも、同じ内容の多数のROM を用意する必要がないので交換する手間が少なく、従ってトラブルが減りメンテナンスが容易である。実際このシステムでは18個のROM が1個で済んだことになる。この方式を使えば、例えば32ビットのCPU を18個使った場合 $4 \times 18 = 72$ 個のROM が4個で済むことになり、その違いは経済的にも大きい。ROM にはモニタ以外にグラフィックルーチンやQSORT のアプリケーションプログラムなどが格納されているが、これらについても同様なメリットがある。

5. 2 機能

モニタは、プログラム実行、1ステップトレース、メモリ内容表示・変更、レジスタ内容表示・変更、ブレークポイント設定・解除、回線端末制御、オブジェクトプログラムロード等のコマンドをサポートする。これらのコマンド機能は随時改良変更している。

また、モニタとは直接関係ないが、共有ROM には各種プログラムが格納されており、必要があれば、プログラムからこれと呼び出し使用することが出来る。

6. 並列処理コンピュータ

このシステムは複数のコンピュータとして複数の学生が同時にアセンブリプログラミングの演習に使用できることは、前章まで述べた通りであるが、1台のマルチプロセッサシステムとしても使用できることは既に明らかである。すなわち、マルチプロセッサシステムとは、個々のCEが協調して全体としてあるまとまった仕事をする並列処理コンピュータのことである。個々のCEが協調しあうためには、CEが相互にデータの交換や連絡が出来なくてはならない。

6. 1 データ交換

このシステムでは、共有メモリ空間に512kバイトの読み書き可能なメモリをもっていることは既にのべたが、その他に、CEが相互に割り込みをかけることができる機能も備えている。この割り込み機能によって、CEが相互に効率よく連絡をとることが出来る。図5を用いてこのメカニズムを説明する。

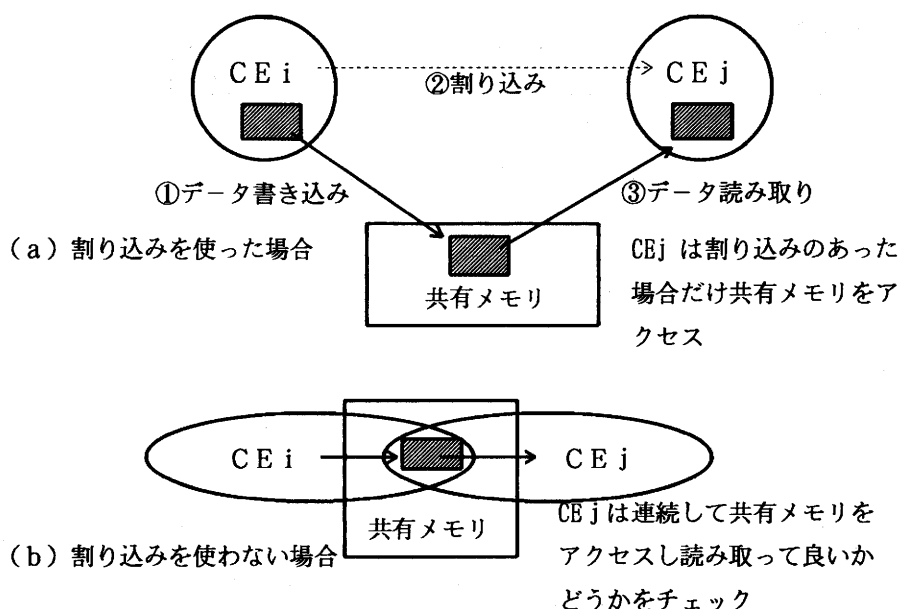


図5 データ交換のメカニズム

図5 (a) は、CE i が処理した共有メモリ上のデータを、CE j が読み取ってそれを自分のデータとして処理する場合の様子を示しているが、この場合CE j が当該のデータを読み取っても良いことを、CE i からCE j への割り込み信号によってCE j が知ることになるためCE j は共有メモリを常にアクセスして見ている必要がない。従って全体の処理の効率も良い。図5 (b) はやはりCE i が処理した共有メモリ上のデータを、CE j が読み取ってそれを自分のデータとして処理する場合の様子を示しているが、この場合CE j が当該のデータを読み取っても良いかどうかを、CE i から共有メモリを通じて教えてもらう必要がある。このためにCE j は共有メモリを常にアクセスしていなければならない。同時にこれは、CE i の共有メモリに対する処理の効率の低下を意味する。

割り込みはハードウェア的に決められたアドレスへ書き込む動作を行うことによって、実現出来る。図6に示すように、IRQAD(\$C8800) から連続した18バイトがCPU0~CPU17 に対する割り込みのアドレスとなっている。これにたいするCPU のベクタ先頭アドレスは\$68であり、\$68番地からの4バイトにこの割り込みに対するルーチンの先頭アドレスをセットしておけばよい。

6. 2 並列処理

並列処理をするには、並列処理を分担する全てのプロセッサにそれぞれプログラムをロードしておかなければならない。図7を使ってこの過程を示す。このシステム自身はプログラムを保存する手段をもたない。CE 0 はWSに作られたSフォーマットと呼ばれる形式のオブジェクトプログラムを共有メモリにロードしたあと、他のCEに割り込みを使ってプログラムの準備が出来たことを知らせる。

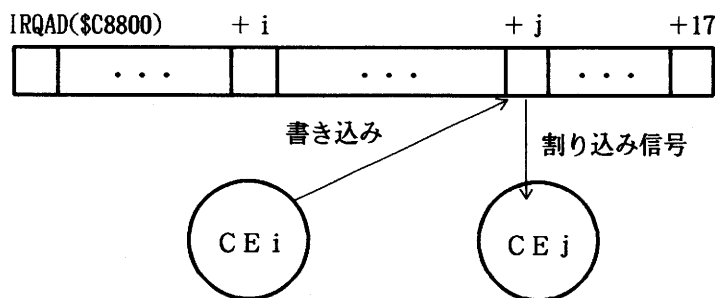


図6 CEへの割り込み

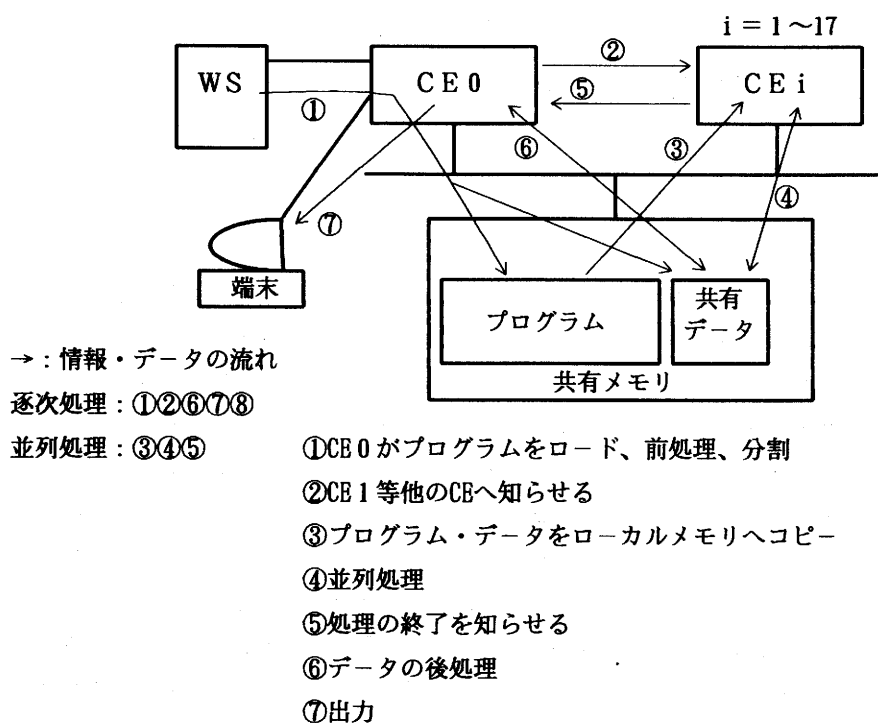


図7 並列処理の進め方

知らせを受けたCE 1 ~ CE 17は共有メモリから必要なプログラムを自分のローカルメモリにコピーしたあと、分担処理するデータがCE 0から分配されるのを待つ。CE 0は必要な前処理（逐次処理）をしたあとデータを分配し、並列処理を指令する（並列処理）。データの処理をおこなっている間にCE相互で同期を取る必要があれば割り込みを使って相互に連絡を取り合い、それぞれ処理を進め、全ての並列処理がすめば、最後に処理を終えたCEがCE 0に割り込みを使って報告する。CE 0は個々に並列処理されたデータをまとめ、さらに必要があれば、後処理をおこない、結果を端末に出力す

る。並列処理システムとして使うときには、CE 1～CE17のモニタは最初は割り込みを受けたらプログラムを共有メモリからロードするなどの簡単な機能しかもたされていない。モニタのROMはスイッチで切替可能となっている。

6. 3 並列処理の例

次に、並列処理を行った例を示す。図8と図9は二つのプログラムについて並列処理をおこなった場合の処理の時間を縦軸に、使ったプロセッサの数を横軸にして関係をグラフで示したものである。図8の場合はCEの数が増えても速度は殆ど上がらない（CEが16で約2.7倍）が、図9の場合は速度の向上（CEが16で約10.3倍）が見られる。

7. 授業において

アセンブリ言語の講義および演習は、1年生のプログラミング学習の後で受けることになるが、ホストとして使用するコンピュータの”設計思想”がまったく異なっており、そのためか学生は随分と苦勞をしているようである。学期末迄にこのシステムを”マルチプロセッサ”として使用できる程度まで力をつける学生は全体の約1割程度にとどまっている。

8. まとめ

アセンブリ言語によるプログラミング技術を初心者に教育する際に、出来るだけ計算機を裸に近い形で多数の学習者に提示すると言う考え方に基き、マイクロプロセッサを使ったボードコンピュータを制作し実際に学習用に使用した。このとき、同じコンピュータを複数台作るなら、あとあと柔軟な使い方が出来ると考えて共有メモリ結合型マルチマイクロコンピュータ（すなわち、個々のCPUが独立して制御できる非同期シリアル回線2組及び、ローカルメモリを持つ）を制作しアセンブリ言語の学習に使用した例について報告した。また、マルチプロセッサシステムとしての評価も試みた。

アセンブリ言語の演習はこのシステムのほかに、WSで実行できるソフトウエアシミュレータも使われて行われ、極めて柔軟な教育システム環境が実現されている。

また今後マルチプロセッサシステムが次第に一般に使われるようになるものと思われるが、そうなったときシーケンシャルな発想でプログラミングをおこなっていたときとは全く異なった考え方ができないと、新しい計算機が使いこなせないことが予想される。そのためにも、実際に並列計算機を使って並列プログラミングの教育をおこなっていくことはますます重要になってくるものと思われる。

参考文献

- 1): Technical Summary 16-Bit Microprocessor With 8-Bit Data Bus 68008, MOTOROLA SEMICONDUCTOR TECHNICAL DATA, MOTOROLA Inc. (1988)
- 2): GAL DATA BOOK, Lattice Semiconductor Corporation (Spring 1988)
- 3) 土井範久: 同期問題, 情報処理, 27, 9, 1022-1030 (1986)
- 4) 坂井修一: 並列計算機におけるスケジューリングと負荷分散, 情報処理, 27, 9, 1031-1040 (1986)

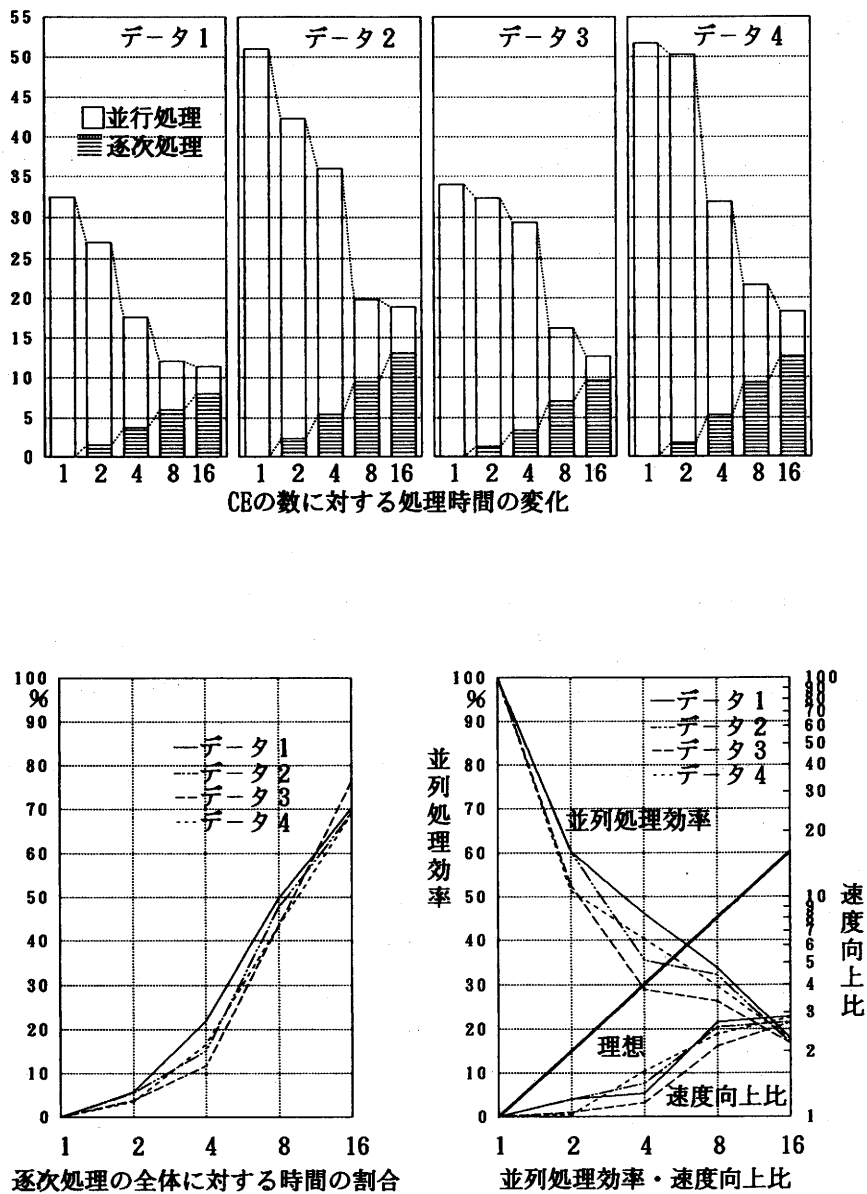


図 8 並列処理の例 (プログラム 1)

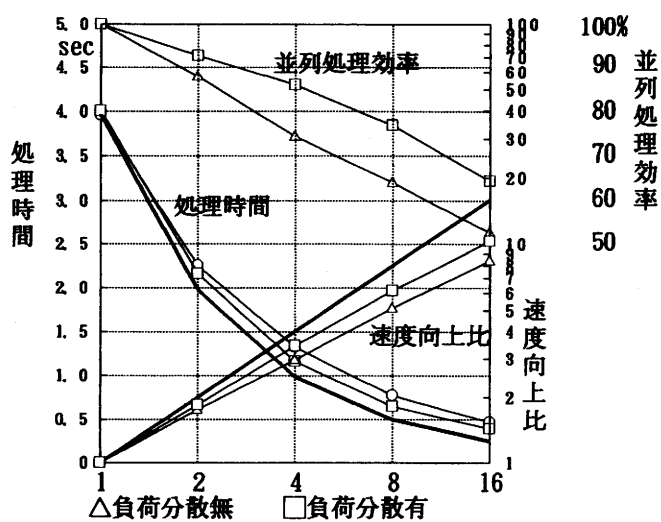


図9 並列処理の例 (プログラム2)

